Documenting and automating your work

Keeping track of your work

- It’s tedious
- It’s important
- It’s a favor to collaborators and to future you

Consistent file and folder naming

General theme: Scale ruins all informality. Think ahead!

- Consider:
  - Project name or acronym
  - Archive or collection information (if applicable)
  - Researcher initials
  - Date (consistently formatted, i.e. YYYYMMDD)
  - Version number (with leading zeros)
- File and folder names should be consistent but unique
  - Quick find-and-sort
- Avoid special characters

Date tip

BAM Co-Exp Run 01 20140904.txt
BAM Co-Exp Run 02 20140904.txt
BAM Co-Exp Run 03 20140904.txt

vs.

Run 1 B anth meth Sept 4 .txt
BAM Rxn 2 2014_09_04.txt
20140904_meth_3.txt

Choosing a controlled vocabulary

- Take the guess work out of choosing between:

Example organization

spring17\university_of_earthish\joe_human_papers\jhp_letters
\jhp_diaries
\jhp_clippings
\jhp_photos

…\...\letters\jhp_box1
\jhp_box2
\jhp_box3

…\...\...\jhp_box1\jhp_1_notes.txt
\jhp_1_meta.csv
\jhp_1_1_1.jpg
\jhp_1_1_2.jpg
\jhp_1_1_3.jpg

Data documentation continuum

README Template

```
README TEMPLATE

Dataset Title:

Author(s):

Description:

Methods and Processing:

Funding Source:

Supplements:
```

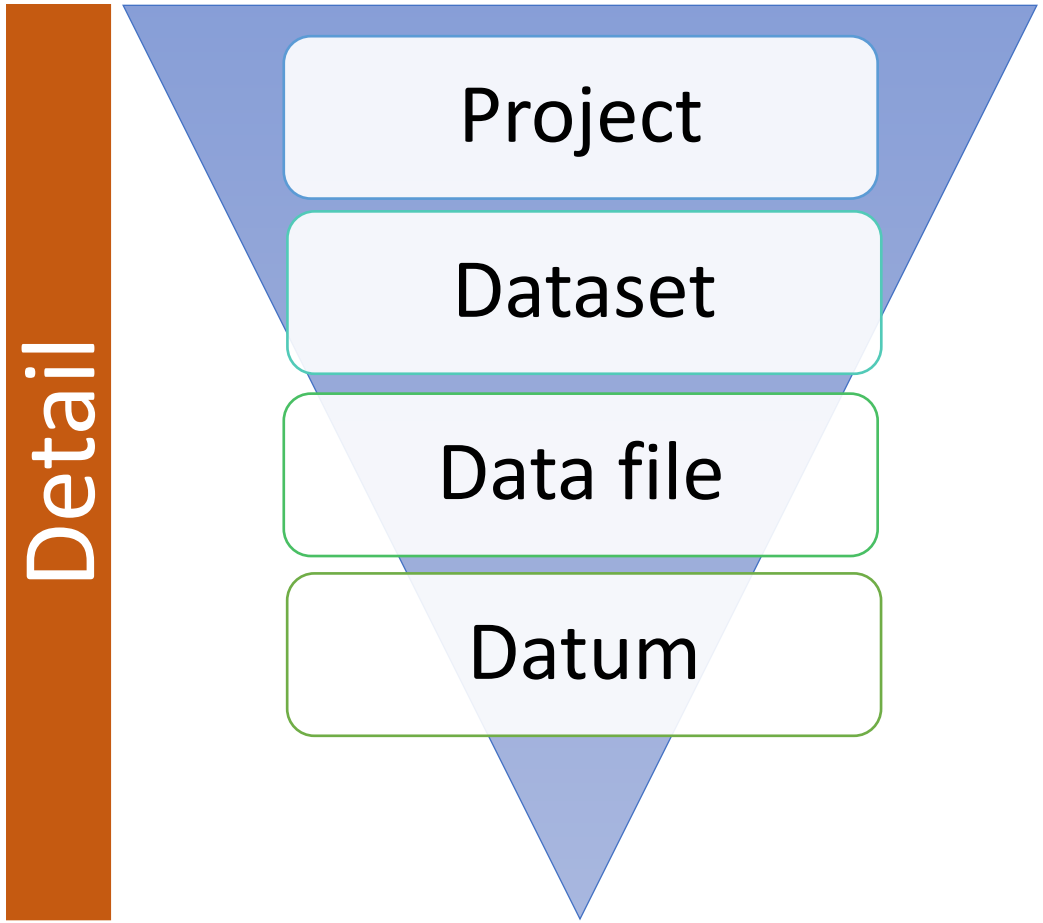
```
<?xml version="1.0"?>

<metadata
  xmlns="http://example.org/myapp/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://example.org/myapp/ http://example.org/myapp/schema.xsd"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <dc:title>
    UKOLN
  </dc:title>
  <dc:description>
    UKOLN is a national focus of expertise in digital information
    management. It provides policy, research and awareness services
    to the UK library, information and cultural heritage communities.
    UKOLN is based at the University of Bath.
  </dc:description>
  <dc:publisher>
    UKOLN, University of Bath
  </dc:publisher>
  <dc:identifier>
    http://www.ukoln.ac.uk/
  </dc:identifier>

</metadata>
```

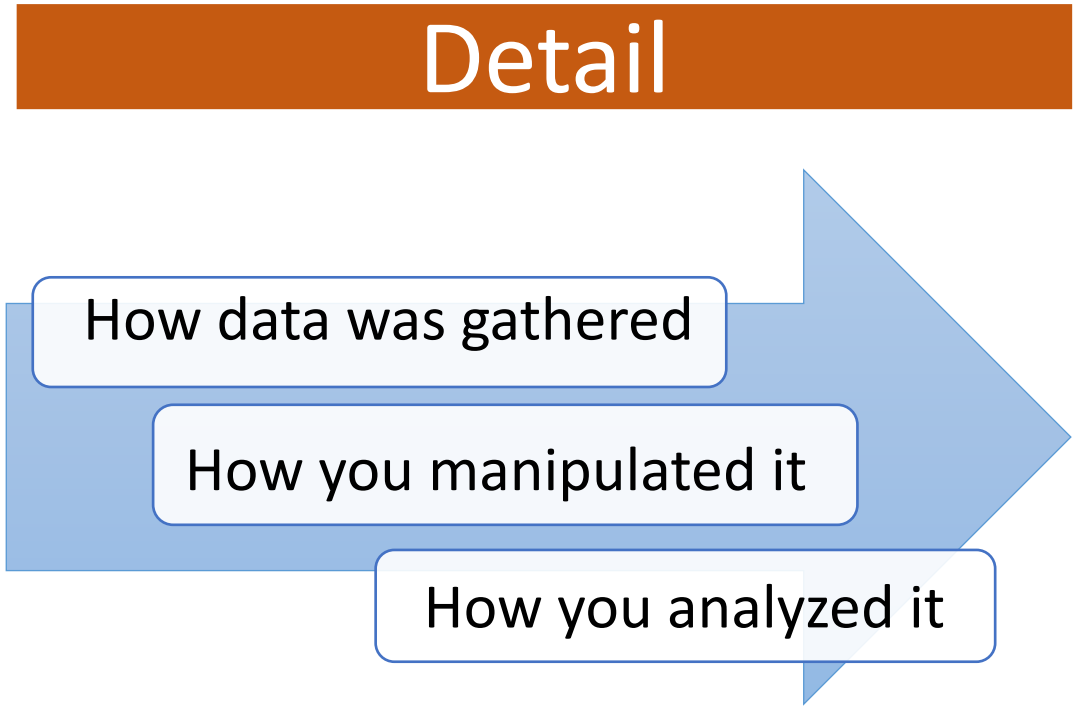
Note that the `http://example.org/myapp/schema.xsd` XML schema does not exist - this is a ficticious example.

Informal ReadMe → **Formal Schema**

Low-Barrier	**Low-Quality**	**High-Quality**	**High-Barrier**
Fast	Irregular	Standardized	Slow
Easy	Incomplete	Rich	Skilled

Documentation Content





Levels to document

- Project
  - What was done, with what tools, to what
- Dataset:
  - Manifest of files
- Data files:
  - Contents and file names
- Data point:
  - Codebook of text content, units

Minimum viable documentation

- Documentation does not need to be:
  - A dissertation
  - Overly detailed
- Documentation should be:
  - Enough information that others can make sense of your data later

Creating metadata

- Store info about your data (the metadata) with your data
- Built in metadata functionality:
  - Equipment: cell phones, cameras, scanners
  - Software: Microsoft Word, Adobe Photoshop
- Common metadata tools:
  - Spreadsheet software
  - Text editors

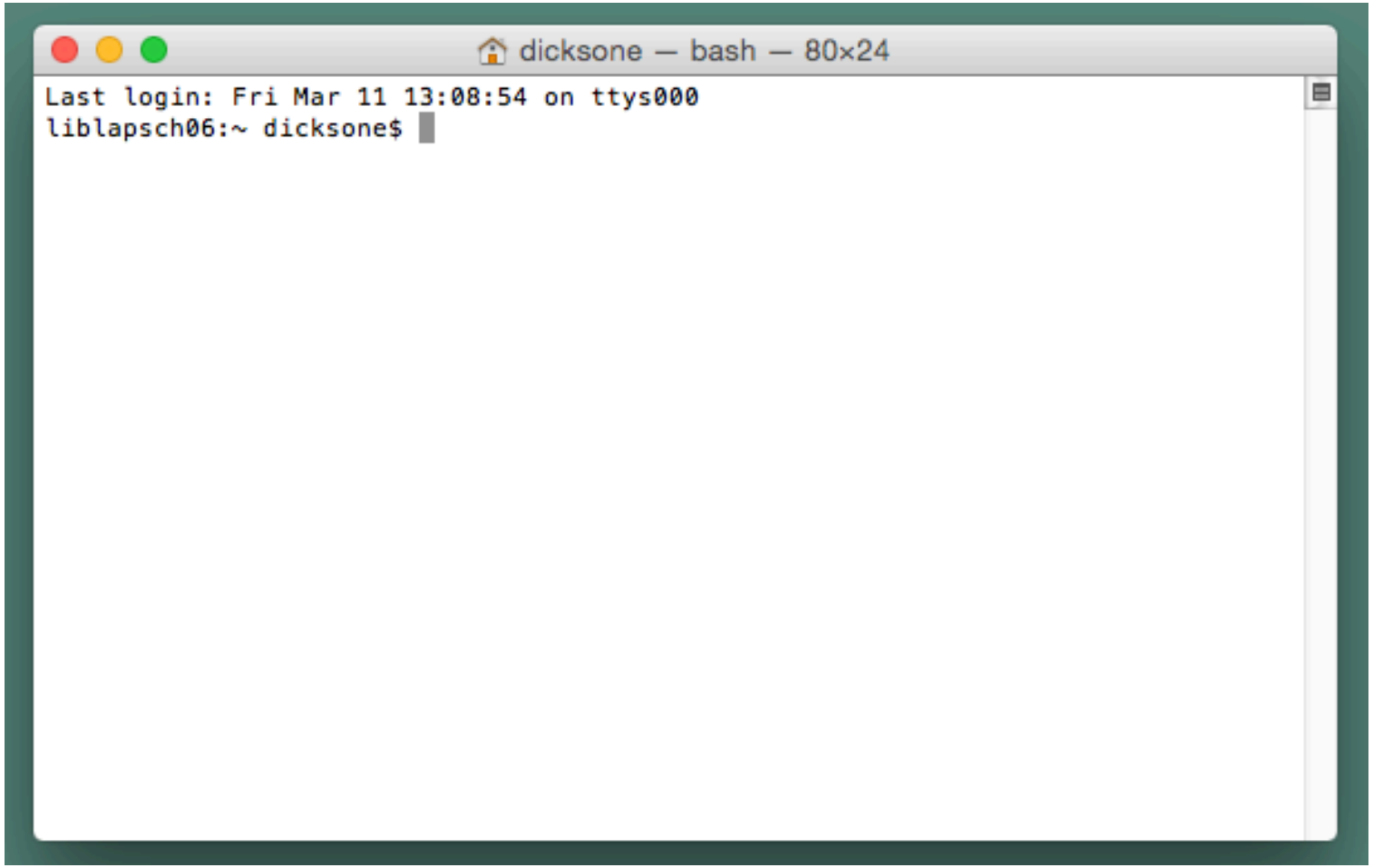
Examples of Documentation

- Readme Files
  - Text files that provides basic information about a dataset, such as:
  - Manifest of files and folders
  - Author, year, associated publication as appropriate
  - Explanation of naming conventions
  - Relationship between directory structure and the data
- Data Dictionaries/Codebooks
  - “Provides a detailed description of each element or variable in your dataset” https://www.dataone.org/best-practices/create-data-dictionary

Annotate your workflow

- Take a few minutes, look at the workflow you created earlier and brainstorm/imagine how you would:
  - Organize the files you will create
  - Name files/folders
  - Document your work

Intro to the command line



Background

- Command line interfaces / Shells
  - On Mac: Terminal
  - On Windows: Command Prompt

Unix shell

- Mac: Terminal
- Windows:
  - Depends on version of Windows
  - Alternatives:
    - Cygwin, a unix-like shell for Windows
    - GitBash
- Bash = Unix shell

Unix

May be Unix

Unix-like

We’re going to use the built-in Bash console environment of Python Anywhere OR Terminal on your Mac

Tips for working in a shell

- Directory = folder
- Case, spaces, and punctuation matter
- Tab to autocomplete a line
- Hit up/down arrow to see last commands entered

Basic Bash commands

- pwd – See which directory you're in

```
pwd
```

- ls – List the files and directories

```
ls -l
```

- mkdir – Make a directory

```
mkdir project1
```

- less – View, but not edit a file; hit "q" to quit viewing

```
less README.txt
```

- mv – Rename a file

```
mv README.txt README1.txt
```

- cd – Change directory

```
cd /home
```

PDFtk

- https://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/
- Command-line tool for working with PDFs
  - Bulk rename
  - Join files
  - Auto-rotate
- Example: Remove page 1 from pdf

```
pdftk awakening_orig.pdf cat 2-12 output
awakening_new.pdf
```

SourceCaster

- https://datapraxis.github.io/sourcecaster/
- Suggested Bash command for working with files (in bulk!)
  - Change file type
  - Change file names
  - Scrape files from the web
- Download the dependencies!
- Example: rename all files ending with .txt extension

```
for file in *.pdf; do mv "$file" "${file/new/}"; done
```

Tesseract

- https://github.com/tesseract-ocr/tesseract
- Command-line Optical Character Recognition tool
- Works with TIFF

Additional resources

- Unclean, unclean! What historians can do about sharing our messy research data
  - https://earlymodernnotes.wordpress.com/2013/05/18/unclean-unclean-what-historians-can-do-about-sharing-our-messy-research-data/
- Embarrassments of Riches: Managing Research Assets
  - http://miriamposner.com/blog/embarrassments-of-riches-managing-research-assets/
- Camera, laptop, and what else?: Hacking better tools for the short archival research trip
  - http://cliotropic.org/blog/talks/camera-laptop-and-what-else/
- Preserving your Research Data
  - http://programminghistorian.org/lessons/preserving-your-research-data